SMTP Client

עד כה למדנו על שני פרוטוקולים נפוצים בשכבת האפליקציה. כעת יש בידיכם את הכלים להבין בעצמכם איך פועלים פרוטוקולים בשכבת האפליקציה. כדי לתרגל זאת נבצע מחקר של פרוטוקול שנקרא SMTP. כל מה שעליכם לדעת בשלב זה, הוא ש-SMTP הוא פרוטוקול שהיה נפוץ מאד בעבר ושימש לשליחת דואר אלקטרוני. את יתר הפרטים אודות הפרוטוקול תסיקו בעצמכם.

בתור התחלה, **תאבחנו** את הפרוטוקול, כלומר תנתחו בעצמכם, מתוך דוגמה של שימוש בפרוטוקול, את הפקודות, השלבים והפרמטרים שלו. לאחר מכן, **תממשו** לקוח SMTP בפייתון שיישלח בעצמו דואר אלקטרוני.

The sniff files are on the same folder as this file and end with .pcap - If wireshark is installed they can be opened by double-click. (smtp1.pcap and smtp2.pcap)

קבצים אלו מכילים הסנפות של מחשב מסויים. בכל הסנפה ישנו משלוח מייל באמצעות פרוטוקול SMTP. מטרתכם היא להבין את פרוטוקול SMTP. כלומר – להבין את מבנה הפרוטוקול, הדרך שבה הוא עובד ואיך יש לרשום בו הודעות על מנת להצליח לשלוח אימיילים באמצעותו. תעשו זאת באמצעות שימוש ב-Wireshark, ניסוי וטעייה, וללא שימוש במקורות חיצוניים. פעולה זו, של הבנת דרך הפעולה מהתבוננות בהסנפות, נקראת "אבחון" והיא שימושית במקרים בהם או צריכים להבין איך עובדת מערכת מבלי לנו תיעוד מלא שלה.

התרכזו ושימו דגש בצד הלקוח של הפרוטוקול (הצד ש**שולח** את המייל), אין צורך להיכנס לפרטים ברמה של השרת.

הנחיה חשובה: אין להשתמש או להיעזר באינטרנט עבור תרגיל זה!

הנחיות נוספות:

- ראשית, הבינו מה היא כתובת ה-IP של המחשב שממנו מתבצעת ההסנפה.
- סננו את הפקטות כך שתראו רק את פקטות של פרוטוקול SMTP.
  - שימו לב: לאחר הסינון ייתכן ותראו גם פקטות מסוג פרוטוקול IMF. התעלמו מהן במהלך התרגיל.
- היעזרו ב-Follow TCP Stream, אותו הכרתם בפרק Wireshark ומודל חמש השכבות /Follow TCP/UDP Stream, כדי לראות את מהלך הפרוטוקול בצורה נוחה.
- נסו להבין - מה הן הפקודות שבפרוטוקול? מה הן הבקשות והתגובות? אילו פרמטרים יש לכל פקודה?

- לאורך התרגיל, זכרו את מטרת פרוטוקול SMTP – לשלוח מיילים. עצרו וחשבו כיצד אתם שולחים דואר אלקטרוני – איזה מידע אתם נדרשים לספק לשם כך? איזה מידע משתנה בין מייל למייל ואיזה מידע לא משתנה? חפשו זאת בקבצי ההסנפה.
- נסו להבין את התמונה הכוללת לפני שאתם צוללים לפרטים. למשל, לו הייתם נדרשים לאבחן את פרוטוקול HTTP, חשוב קודם להבין שהמבנה הבסיסי הוא בקשה-תגובה, וכי יש סוגי בקשות שונים כמו GET ו-POST, הרבה לפני שצוללים לסוגי ה-Headers השונים.
- במהלך התרגיל, תצטרכו להשתמש בקידוד Base64. את הקידוד הזה פגשנו באותנטיקציה של HTTP, ואז השתמשנו באתר אינטרנט כדי להמיר אל הקידוד ובחזרה. הפעם נשתמש בפייתון.

על מנת לקודד מחרוזת לקידוד Base64 באמצעות פייתון, נשתמש בפעולה **encode**:

```
>>> my_string = ‘This is a string...’
>>> my_string.enode(‘base64’)
‘VGhpcyBpcyBhIHN0cmluZW==\n’
```

שימו לב כי פייתון מוסיף למחרוזת את '\n' (התו של ירידת שורה) שאינו חלק מקידוד Base64, ולכן עליכם להסירו. סימני השווה (=) הם כן חלק מקידוד Base64.

על מנת לבצע את הפעולה ההפוכה, נשתמש בפעולה **decode**:

```
>>> ‘VGhpcyBpcyBhIHN0cmluZW==\’.decode(‘base64’)
‘This is a string...’
```

בתרגיל זה תממשו בעזרת פייתון לקוח שישלח דואר אלקטרוני באמצעות פרוטוקול SMTP.

הנחיה חשובה: יש להשתמש בספריית socket בלבד. אין להשתמש בספריות עזר כגון smtplib.

על מנת לעשות זאת, השתמשו במדמה שרת SMTP, שנמצא בכתובת: networks.cyber.org.il. השרת מאזין על פורט סטנדרטי של SMTP, שמספרו 587.

מדמה השרת של גבהים יקבל את המידע שנשלח מהלקוח שלכם. אם המידע התואם את דרישות הפרוטוקול, השרת יחזיר לכם תשובה לפי הפרוטוקול וימתין לבקשה הבאה. אם הלקוח שלכם שגה במימוש הפרוטוקול, שרת גבהים יחזיר תשובה "הבקשה אינה לפי הפרוטוקול" וינתק את ההתקשרות.

אם מימשתם את הפרוטוקול נכון מתחילתו ועד סופו, שרת גבהים יחמיא לכם על סיום התרגיל בהצלחה.

דגשים למימוש:

- השרת מקפיד על כך שכל הודעה שנשלחת אליו תתאים להודעה שבפקטות שבקובץ ההסנפה.

- שימו לב לצירופי תווים מיוחדים, אם צריכים להיות כאלה, בסוף כל הודעה.
- השרת ניתק אתכם? תקנו את הבעיה והתחברו שוב.
- השרת אינו בודק שהסיסמה ושם המשתמש שלכם מתאימים. במילים אחרות - אפשר להתחבר עם כל שם משתמש וסיסמה.
- השרת אינו מתיימר לממש את כל הפקודות של פרוטוקול SMTP, אלא רק לוודא שהצלחתם לאבחן את הפרוטוקול בצורה מוצלחת.
- כמובן שהשרת גם אינו שולח את המייל ששלחתם ☺ זהו תרגיל אבחוני בלבד.

הצלחה והנאה בתרגיל זה!